EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	18	chih$1hung near chien.in.	US-PGPUB; USPAT	OR	ON	2007/10/02 10:41
S2	194	mitac.as.	US-PGPUB; USPAT	OR	ON	2007/10/02 10:41
S3	0	S2 and (reference and pointer).clm.	US-PGPUB; USPAT	OR	ON	2007/10/02 10:42
S4	10	S2 and (reference).clm.	US-PGPUB; USPAT	OR	ON	2007/10/02 10:42
S5	1056	717/120-123.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:43
S6	16	S5 and (reference adj code)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:43
S7	66	(source adj code) with insert$4 with reference	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:58
S8	55	S7 and (@pd<"20030827" or @ad<"20030827" or @prad<"20030827" or @rlad<"20030827")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:10
S9	1266	programming adj reference	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:55
S10	22	S9 and (reference near code)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:55

EAST Search History

S11	26	(source adj code) with insert$4 with pointer	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 10:59
S12	8	("5280617" | "5519868" | "5632035" | "5740443" | "5790867" | "5835771" | "6041180" | "6141792").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/10/02 11:12
S13	6	("5642514" | "5790867" | "5812854").PN. OR ("6041180"). URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/10/02 11:37
S14	878	(updat$4 or modify$4) near (source adj code)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 11:42
S15	6	S14 and (insert$4 near reference)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 11:42
S16	7	("5495613" | "5511159" | "5778371" | "6321372").PN. OR ("6802059").URPN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/10/02 11:45
S17	1003	link$4 near3 (source adj code)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 11:47
S18	3	S17 and (insert$4 near3 (reference near code))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 11:48
S19	4613	717/140-167.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:04

EAST Search History

S20	3	S19 and (insert$4 near3 (reference near code))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:07
S21	251	S19 and (reference near code)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:08
S22	141	S21 and insert$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:10
S23	124	S22 and (@pd<"20030827" or @ad<"20030827" or @prad<"20030827" or @rlad<"20030827")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:10
S25	2	"5649204".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/02 12:30
S26	15	("5649204").URPN.	USPAT	OR	ON	2007/10/02 12:31
S27	11	("4558413" | "4672532" | "4809170" | "4941084" | "4987531" | "5065400" | "5109484" | "5109486" | "5175828" | "5237688" | "5247679").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2007/10/02 14:18

EAST Search History

S28	50	("20050050394" "4999554" "5285376" "5379389" "5764991" "5784605" "6034730" "6071317" "6098111" "6308171" "6336209" "6665735" "6002867" "20050021754" "20050066317" "5481294" "4380006" "4860114" "5297023" "5396631" "5404588" "5446852" "5748963" "5819235" "6035282" "6138073" "6138059" "6144889" "6243864" "6415302" "6424977" "6434576" "6434577" "6449626" "6999980" "4245323" "4480280" "4513677" "4535333" "4631678" "4633384" "4773041" "4780809" "4791405" "4812924" "4841132" "4860378" "4864530" "4873720" "4893291").pn.	US-PGPUB; USPAT	OR	ON	2007/10/02 14:45

Terms used: **inserting reference code**

Found **80,013** of **212,128**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

21 Incremental dependence analysis for interactive parallelization

Kevin Smith, Bill Appelbe, Kurt Stirewalt

June 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 4th international conference on Supercomputing ICS '90**, Volume 18 Issue 3b

Publisher: ACM Press

Full text available: pdf(1.20 MB) Additional Information: full citation, abstract, references, citings, index terms

Incrementally updating dependence information during interactive parallelization is a difficult proposition. We have developed a tool (PAT) that maintains dependence information during incremental transformations to a Fortran program, including loop parallelization, code replication, alignment and shifting, as well as insertion and deletion of code including parallel primitives. Our analysis is based on a variation on the s ...

22 Efficient simulation of cache memories

S. Dwarkadas, J. R. Jump, J. B. Sinclair

October 1989 **Proceedings of the 21st conference on Winter simulation WSC '89**

Publisher: ACM Press

Full text available: pdf(930.36 KB) Additional Information: full citation, abstract, references, citings, index terms

Cache memories are used in computer systems to reduce average memory access times. Existing techniques for predicting cache performance are often unsatisfactory in terms of cost or performance. This paper presents a method for efficiently simulating the effects of a cache on the execution time of a program. We use an execution-driven simulation approach that requires no hardware support and provides a highly accurate dynamic address trace to a cache simulation model. Almost all of the overhead i ...

23 Heap profiling for space-efficient Java

Ran Shaham, Elliot K. Kolodner, Mooly Sagiv

May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: pdf(1.52 MB) Additional Information: full citation, abstract, references, citings, index terms

We present a heap-profiling tool for exploring the potential for space savings in Java programs. The output of the tool is used to direct rewriting of application source code in a way that allows more timely garbage collection (GC) of objects, thus saving space. The rewriting can also avoid allocating some objects that are never used.

The tool measures the difference between the actual collection time and the potential

earliest collection time of objects for a Java application. This t ...

24 Timing analysis: Reducing program image size by extracting frozen code and data

Daniel Citron, Gadi Haber, Roy Levin
September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04**
Publisher: ACM Press
Full text available: pdf(167.75 KB) Additional Information: full citation, abstract, references, index terms

Constraints on the memory size of embedded systems require reducing the image size of executing programs. Common techniques include code compression and reduced instruction sets. We propose a novel technique that eliminates large portions of the executable image without compromising execution time (due to decompression) or code generation (due to reduced instruction sets). *Frozen* code and data portions are identified using profiling techniques and removed from the loadable image. They are ...

Keywords: feedback directed, frozen code, frozen data, image size

25 Session S4.2: program transformation: Cost effective memory disambiguation for multimedia codes

Esther Salamí, Jesús Corbal, Carlos Álvarez, Mateo Valero
October 2002 **Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems CASES '02**
Publisher: ACM Press
Full text available: pdf(242.97 KB) Additional Information: full citation, abstract, references, citings, index terms

Frequently, ambiguous memory references prevent the compiler from exploiting all the available ILP. Techniques to detect aliasing between access patterns of array elements are quite effective for many numeric applications, but although media codes usually process disjointed streams that exhibit regular access patterns, current commercial compilers remain unsuccessful in disambiguating them due mainly to complex pointer references. In this paper we propose a very cost effective disambiguation met ...

Keywords: VLIW, memory disambiguation, multimedia, run-time analysis, time-to-market

26 Feedback linking: optimizing object code layout for updates

Carl von Platen, Johan Eker
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN/SIGBED conference on Language, compilers and tool support for embedded systems LCTES '06**, Volume 41 Issue 7
Publisher: ACM Press
Full text available: pdf(421.85 KB) Additional Information: full citation, abstract, references, index terms

Firmware over the air (FOTA) is becoming a standard procedure for maintaining and updating wireless embedded systems. To cope with bandwidth and storage constraints this is facilitated using incremental updates based on delta technology, i.e. only the modifications are transmitted. The performance of a FOTA update is highly dependent on the size of the delta, and the type of modifications. Application of a delta update involves mutating the present version, byte by byte, into the new version. Th ...

Keywords: flash memory, incremental software update

27 Managing bounded code caches in dynamic binary optimization systems

Kim Hazelwood, Michael D. Smith
September 2006 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 3 Issue 3

Publisher: ACM Press
Full text available: pdf(666.72 KB) Additional Information: full citation, abstract, references, index terms

Dynamic binary optimizers store altered copies of original program instructions in software-managed code caches in order to maximize reuse of transformed code. Code caches store code blocks that may vary in size, reference other code blocks, and carry a high replacement overhead. These unique constraints reduce the effectiveness of conventional cache management policies. Our work directly addresses these unique constraints and presents several contributions to the code-cache management problem. ...

Keywords: Dynamic optimization, code caches, dynamic translation, just-in-time compilation

28 The Performance of Runtime Data Cache Prefetching in a Dynamic Optimization System
Jiwei Lu, Howard Chen, Rao Fu, Wei-Chung Hsu, Bobbie Othmer, Pen-Chung Yew, Dong-Yuan Chen
December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture MICRO 36**
Publisher: IEEE Computer Society
Full text available: pdf(253.79 KB) Additional Information: full citation, abstract, citings, index terms

Traditional software controlled data cache prefetching isoften ineffective due to the lack of runtime cache miss andmiss address information. To overcome this limitation, weimplement runtime data cache prefetching in the dynamicoptimization system ADORE (ADaptive Object code RE-optimization).Its performance has been compared withstatic software prefetching on the SPEC2000 benchmarksuite. Runtime cache prefetching shows better performance.On an Itanium 2 based Linux workstation, it can increasepe ...

29 Compile-time memory reuse in logic programming languages through update in place
Gudjón Gudjónsson, William H. Winsborough
May 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 21 Issue 3
Publisher: ACM Press
Full text available: pdf(693.38 KB) Additional Information: full citation, abstract, references, index terms

Standard implementation techniques for single-assignment languages modify a data structure without destroying the original, which may subsequently be accessed. Instead a variant structure is created by using newly allocated cells to represent the changed portion and to replace any cell that references a newly allocated cell. The rest of the original structure is shared by the variant. The effort required to leave the original uncorrupted is unnecessary when the program will never reference ...

Keywords: Prolog, compile-time garbage collection, local reuse, reuse map, update in place

30 Static single assignment form for machine code
Allen Leung, Lal George
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation PLDI '99**, Volume 34 Issue 5
Publisher: ACM Press
Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, citings, index terms

Static Single Assignment (SSA) is an effective intermediate representation in optimizing compilers. However, traditional SSA form and optimizations are not applicable to programs represented as native machine instructions because the use of dedicated registers imposed by calling conventions, the runtime system, and target architecture

must be made explicit. We present a simple scheme for converting between programs in machine code and in SSA, such that references to dedicated physical registers ...

31 Generation and analysis of very long address traces

Anita Borg, R. E. Kessler, David W. Wall
May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture ISCA '90**, Volume 18 Issue 3a
Publisher: ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings, index terms

Existing methods of generating and analyzing traces suffer from a variety of limitations including complexity, inaccuracy, short length, inflexibility, or applicability only to CISC machines. We use a trace generation mechanism based on link-time code modification which is simple to use, generates accurate long traces of multi-user programs, runs on a RISC machine, and can be flexibly controlled. On-the-fly analysis of the traces allows us to get accurate performance data for large second-l ...

32 A systematic approach to advanced debugging: incremental compilation

Peter Fritzson
March 1983 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the symposium on High-level debugging SIGSOFT '83**, Volume 8 , 18 Issue 4 , 8
Publisher: ACM Press
Full text available: pdf(664.22 KB) Additional Information: full citation, abstract, references

This paper presents two topics: Implementation of a debugger through use of an incremental compiler, and techniques for fine-grained incremental compilation. Both the debugger and the compiler are components of the highly Integrated programming environment DICE (Distributed Incremental Compiling Environment) which alms at providing programmer support in the case where the programming environment resides in a host computer and the program. Is running on a target computer that is connected to the ...

33 Reducing the impact of software prefetching on register pressure

David W. Shrewsbury, Cindy Norris
March 2000 **Proceedings of the 2000 ACM symposium on Applied computing - Volume 2 SAC '00**
Publisher: ACM Press
Full text available: pdf(492.51 KB) Additional Information: full citation, references, citings, index terms

34 Multidimensional access methods

Volker Gaede, Oliver Günther
June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2
Publisher: ACM Press

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, citings, index terms

Search operations in databases require special support at the physical level. This is true for conventional databases as well as spatial databases, where typical search operations include the point query (find all objects that contain a given search point) and the region query (find all objects that overlap a given search region). More than ten years of spatial database research have resulted in a great variety of multidimensional access methods to support ...

Keywords: data structures, multidimensional access methods

35 Power: A compiler approach for reducing data cache energy

W. Zhang, M. Karakoy, M. Kandemir, G. Chen

June 2003 **Proceedings of the 17th annual international conference on Supercomputing ICS '03**

Publisher: ACM Press

Full text available: pdf(299.23 KB) Additional Information: full citation, abstract, references, citings, index terms

Silicon technology advances have made it possible to pack millions of transistors --- switching at high clock speeds --- on a single chip. While these advances bring unprecedented performance to electronic products, they pose difficult power/energy consumption problems. For example, large number of transistors in dense on-chip cache memories consume significant static (leakage) power even if the cache is not used by the current computation. While previous compiler research studied code and data ...

Keywords: compiler analysis, data caches, energy optimization

36 Compiler algorithms for event variable synchronization

Zhiyuan Li

June 1991 **Proceedings of the 5th international conference on Supercomputing ICS '91**

Publisher: ACM Press

Full text available: pdf(1.00 MB) Additional Information: full citation, references, citings, index terms

37 Automatic incremental state saving

Darrin West, Kiran Panesar

July 1996 **ACM SIGSIM Simulation Digest , Proceedings of the tenth workshop on Parallel and distributed simulation PADS '96**, Volume 26 Issue 1

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(870.62 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

We present an Incremental State Saving technique for which the state saving calls are inserted automatically by directly editing the application executable. This method has the advantage of being easy to use since it is fully automatic, and has good performance since it adds overhead only where state is being modified. Since the editing happens on executable code, the method is independent of the compiler, and allows third party libraries to be used. None of the previous incremental state saving ...

Keywords: Parallel Discrete Event Simulation, State Saving, Incremental State Saving, Checkpointing, Time Warp

38 Register allocation over the program dependence graph

Cindy Norris, Lori L. Pollock

June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation PLDI '94**, Volume 29 Issue 6

Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes RAP, a Register Allocator that allocates registers over the Program Dependence Graph (PDG) representation of a program in a hierarchical manner. The PDG program representation has been used successfully for scalar optimizations, the detection and improvement of parallelism for vector machines, multiple processor machines, and machines that exhibit instruction level parallelism, as well as debugging, the integration of different versions of a program, and translation of ...

39 SafeTSA: a type safe and referentially secure mobile-code representation based on static single assignment form

Wolfram Amme, Niall Dalton, Jeffery von Ronne, Michael Franz

May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, references, citings, index terms

40 Compiler techniques for code compaction

Saumya K. Debray, William Evans, Robert Muth, Bjorn De Sutter

March 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 2

Publisher: ACM Press

Full text available: pdf(409.20 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In recent years there has been an increasing trend toward the incorpor ation of computers into a variety of devices where the amount of memory available is limited. This makes it desirable to try to reduce the size of applications where possible. This article explores the use of compiler techniques to accomplish code compaction to yield smaller executables. The main contribution of this article is to show that careful, aggressive, interprocedural optimization, together with procedural abstr ...

Keywords: code compaction, code compression, code size reduction

Terms used: **header files**

Found **39,329** of **212,128**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

21 A relation-based language interpreter for a content addressable file store

T. R. Addis

June 1982 **ACM Transactions on Database Systems (TODS)**, Volume 7 Issue 2

Publisher: ACM Press

Full text available: pdf(2.50 MB) Additional Information: full citation, abstract, references, citings, index terms

The combination of the Content Addressable File Store (CAFS®; CAFS is a registered trademark of International Computers Limited) and an extension of relational analysis is described. This combination allows a simple and compact implementation of a database query and update language (FIDL). The language has one of the important properties of a "natural" language interface by using a "world model" derived from the relational analysis. The interpreter (FLIN) takes f ...

Keywords: content addressing

22 Managing C++ libraries

J. M. Coggins, G. Bollella

June 1989 **ACM SIGPLAN Notices**, Volume 24 Issue 6

Publisher: ACM Press

Full text available: pdf(859.89 KB) Additional Information: full citation, abstract, citings, index terms

This paper describes a scheme we have used to manage a large library written in the C++ language. The scheme imposes a directory structure, and represents dependency hierarchy in a globally accessible file we call the 'prelude' file. We also discuss the structure of the description files (makefiles) used with the UNIX options we have found to be useful in reducing the size of the library, and how to minimize recompilation time after trivial changes to the source code of the library.

23 Parallel netCDF: A High-Performance Scientific I/O Interface

Jianwei Li, Wei-keng Liao, Alok Choudhary, Robert Ross, Rajeev Thakur, William Gropp, Rob Latham, Andrew Siegel, Brad Gallagher, Michael Zingale

November 2003 **Proceedings of the 2003 ACM/IEEE conference on Supercomputing SC '03**

Publisher: IEEE Computer Society

Full text available: pdf(162.35 KB) Additional Information: full citation, abstract

Dataset storage, exchange, and access play a critical role in scientific applications. For such purposes netCDF serves as a portable, efficient file format and programming interface, which is popular in numerous scientific application domains. However, the original interface does not provide an efficient mechanism for parallel data storage and

access. In this work, we present a new parallel interface for writing and reading netCDF datasets. This interface is derived with minimal changes from the ...

24 PCCTS reference manual: version 1.00

T. J. Parr, H. G. Dietz, W. E. Cohen
February 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 2

Publisher: ACM Press

Full text available: pdf(3.77 MB) Additional Information: full citation, citings, index terms

25 Increasing web server throughput with network interface data caching

Hyong-youb Kim, Vijay S. Pai, Scott Rixner
October 2002 **ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 30 , 37 , 36 Issue 5 , 10 , 5

Publisher: ACM Press

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings

This paper introduces network interface data caching, a new technique to reduce local interconnect traffic on networking servers by caching frequently-requested content on a programmable network interface. The operating system on the host CPU determines which data to store in the cache and for which packets it should use data from the cache. To facilitate data reuse across multiple packets and connections, the cache only stores application-level response content (such as HTTP data), with applica ...

26 Large-scale software development with the Ada Language System

Richard M. Thall
January 1983 **Proceedings of the 1983 computer science conference CSC-83**

Publisher: ACM Press

Full text available: pdf(1.02 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper identifies three major characteristics of large-scale computer programming projects. The design features of the Ada Language System which facilitate large-scale efforts are then described in terms of these characteristics. The Ada Language System is a programming support environment for the Ada Language.

27 A language-based design for portable data files

C. Burch
May 1989 **ACM SIGPLAN Notices**, Volume 24 Issue 5

Publisher: ACM Press

Full text available: pdf(950.29 KB) Additional Information: full citation, abstract, index terms

Currently data files to be accessed remotely from dissimilar systems must be transformed to the local language processors' file format and data representation; a process that has changed little since punch cards were the main form of portable data files. A standard is proposed for languages to use the data type information available to the runtime library to make these data transformations before the data is transferred to the file or the user's variables. By specifying one binary format for eac ...

28 The design and implementation of tripwire: a file system integrity checker

Gene H. Kim, Eugene H. Spafford
November 1994 **Proceedings of the 2nd ACM Conference on Computer and communications security CCS '94**

Publisher: ACM Press

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

At the heart of most computer systems is a file system. The file system contains user data, executable programs, configuration and authorization information, and (usually) the base executable version of the operating system itself. The ability to monitor file systems for unauthorized or unexpected changes gives system administrators valuable data for protecting and maintaining their systems. However, in environments of many networked heterogeneous platforms with different policies and softw ...

29 SynRGen: an extensible file reference generator

Maria R. Ebling, M. Satyanarayanan
May 1994 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1994 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '94**, Volume 22 Issue 1
Publisher: ACM Press

Full text available: pdf(922.24 KB) Additional Information: full citation, abstract, references, citings, index terms

SynRGen, a synthetic file reference generator operating at the system call level, is capable of modeling a wide variety of usage environments. It achieves realism through trace-inspired micromodels and flexibility by combining these micromodels stochastically. A micromodel is a parameterized piece of code that captures the distinctive signature of an application. We have used SynRGen extensively for stress testing the Coda File System. We have also performed a controlled ex ...

30 Reading file metadata with extract and libextractor

Christian Grothoff
June 2005 **Linux Journal**, Volume 2005 Issue 134
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(14.86 KB) Additional Information: full citation, abstract, index terms

Where are the 400x200 PNG images I worked on in March? This system offers the answer.

31 Storage protocol designs: NFS over RDMA

Brent Callaghan, Theresa Lingutla-Raj, Alex Chiu, Peter Staubach, Omer Asad
August 2003 **Proceedings of the ACM SIGCOMM workshop on Network-I/O convergence: experience, lessons, implications NICELI '03**
Publisher: ACM Press
Full text available: pdf(126.79 KB) Additional Information: full citation, abstract, references

The NFS filesystem was designed as a work-group filesystem, making a central file store available to and shared between a number of client workstations. However, more recently NFS has grown in popularity in the server room, connecting large application servers with back-end file servers. In this environment, where high-speed access to data is critical, high capacity interconnects like gigabit Ethernet, Fibre Channel and Infiniband are to be expected. With RDMA technology we can fully utilize the ...

32 Industrial Session: Scalable streaming of JPEG2000 images using hypertext transfer protocol

Sachin Deshpande, Wenjun Zeng
October 2001 **Proceedings of the ninth ACM international conference on Multimedia MULTIMEDIA '01**
Publisher: ACM Press
Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, index terms

This paper describes a scalable architecture for streaming of JPEG2000 images, using Hypertext Transfer Protocol (HTTP). JPEG2000 is a new image compression standard. One of the goals of JPEG2000 is to support large images. For a large image, even the compressed image file size can be very big. Thus downloading the entire image at its full resolution can take a long time depending upon the user's connection speed. Thus we propose to use streaming of JPEG2000 images. We use Hypertext transfer pro ...

Keywords: HTTP streaming, JPEG2000, image streaming, quality scalability, region-of-interest scalability, resolution scalability, scalable streaming

33 IRON file systems

Vijayan Prabhakaran, Lakshmi N. Bairavasundaram, Nitin Agrawal, Haryadi S. Gunawi, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: pdf(323.82 KB) Additional Information: full citation, abstract, references, citings, index terms

Commodity file systems trust disks to either work or fail completely, yet modern disks exhibit more complex failure modes. We suggest a new *fail-partial failure model* for disks, which incorporates realistic localized faults such as latent sector errors and block corruption. We then develop and apply a novel *failure-policy fingerprinting* framework, to investigate how commodity file systems react to a range of more realistic disk failures. We classify their failure policies in a new ...

Keywords: IRON file systems, block corruption, disks, fail-partial failure model, fault tolerance, internal, latent sector errors, redundancy, reliability, storage

34 Contributions: The AGF plotfile: towards a standardization for storage and transportation of graphics information

G. Enderle, I. Giese, M. Krause, H. P. Meinzer

December 1978 **ACM SIGGRAPH Computer Graphics**, Volume 12 Issue 4

Publisher: ACM Press

Full text available: pdf(836.38 KB) Additional Information: full citation, abstract, references

The AGF plotfile is a sequential file for storage and transportation of graphics information. It contains records describing graphics primitives (points, vectors, polylines, texts) and their attributes. The primitives may be two- or three-dimensional. Raster graphics records are included in the plotfile. The structure of the plotfile is described, the capabilities of an interpreter are outlined, and it is shown how the plotfile can be interfaced to the proposed SIGGRAPH-CORE graphics system.

35 Context-specific middleware specialization techniques for optimizing software product-line architectures

Arvind S. Krishna, Aniruddha S. Gokhale, Douglas C. Schmidt

April 2006 **ACM SIGOPS Operating Systems Review , Proceedings of the 2006 EuroSys conference EuroSys '06**, Volume 40 Issue 4

Publisher: ACM Press

Full text available: pdf(676.55 KB) Additional Information: full citation, abstract, references, index terms

Product-line architectures (PLAs) are an emerging paradigm for developing software families for distributed real-time and embedded (DRE) systems by customizing reusable artifacts, rather than hand-crafting software from scratch. To reduce the effort of developing software PLAs and product variants for DRE systems, developers are applying general-purpose -- ideally standard -- middleware platforms whose reusable services and mechanisms support a range of application quality of service (QoS) requi ...

Keywords: middleware, product lines, specializations

36 Template instantiation for C++

Glen McCluskey, Robert B. Murray

December 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 12

Publisher: ACM Press
Full text available: pdf(693.62 KB) Additional Information: full citation, citings, index terms

37 File system usage in Windows NT 4.0
Werner Vogels
December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP '99**, Volume 33 Issue 5
Publisher: ACM Press
Full text available: pdf(1.75 MB) Additional Information: full citation, abstract, references, citings, index terms

We have performed a study of the usage of the Windows NT File System through long-term kernel tracing. Our goal was to provide a new data point with respect to the 1985 and 1991 trace-based File System studies, to investigate the usage details of the Windows NT file system architecture, and to study the overall statistical behavior of the usage data.In this paper we report on these issues through a detailed comparison with the older traces, through details on the operational characteristics and ...

38 Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++
Andreas Griewank, David Juedes, Jean Utke
June 1996 **ACM Transactions on Mathematical Software (TOMS)**, Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(494.33 KB) Additional Information: full citation, appendices and supplements, abstract, references, cited by, index terms

The C++ package ADOL-C described here facilitates the evaluation of first and higher derivatives of vector functions that are defined by computer programs written in C or C++. The resulting derivative evaluation routines may be called from C/C++, Fortran, or any other language that can be linked with C. The numerical values of derivative vectors are obtained free of truncation errors at a small multiple of the run-time and randomly accessed memory of the given function evaluation program. D ...

Keywords: Hessians, Taylor coefficients, automatic differentiation, chain rule, forward mode, gradients, overloading, reverse mode

39 A survey of computer graphics image encoding and storage formats
Wayne E. Carlson
April 1991 **ACM SIGGRAPH Computer Graphics**, Volume 25 Issue 2
Publisher: ACM Press
Full text available: pdf(900.93 KB) Additional Information: full citation, abstract, index terms

This paper is a survey of several storage formats used for computer generated or sampled images, both for purposes of archival storage and transmission (transfer from one location or platform to another). Various methodologies for the compression of such images are presented and discussed, Image storage standards and some of the more common commercial image storage techniques are presented in terms of the underlying compression algorithms they are based upon and the general internal data structu ...

40 Kernel Korner: The ELF Object File Format: Introduction
Eric Youngdale
April 1995 **Linux Journal**
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(18.79 KB) Additional Information: full citation, index terms

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next